Amendments to the Specification:

Please add the following new paragraph after paragraph [0014]:

[0014.1] Fig.4 is a block diagram of a peripheral device according to another embodiment of the present invention.

Please replace paragraph [0029] with the following amended paragraph:

[0029] In order to prevent the peripheral device 50 from entering the waiting state, the preferred embodiment has to adequately determine capacities of the buffers 58, 60 according to the transfer rate between the controller 52 and the bus interface circuit 54 and the data transfer rate between the bus interface circuit 54 and the bus 56. In addition, the bus interface circuit 54 in the preferred embodiment only has two buffers 58, 60. However, a plurality of buffers can be applied on the bus interface circuit 54 to complete the data transaction. Please refer to Fig.4, which is a block diagram of a peripheral device 150 according to another embodiment of the present invention. Differing from the peripheral device 50 shown in Fig.2, the peripheral device 150 has a bus interface circuit 154 containing three buffers 58, 59, 60. All other features are the same, and the same components are indicated by the same reference numbers. ~~For instance, suppose the bus interface circuit 54 includes three buffers A, B, C.~~ Based on the above-mentioned operations, the bus interface circuit 54 successively activates these buffers 58, 59, 60 ~~A, B, C~~ to store data delivered from the controller 52.

Please replace paragraph [0030] with the following amended paragraph:

[0030] In the beginning, the bus interface circuit 54 enables the ~~buffer A~~ buffer 58 to store data outputted from the controller 52 until the buffer 58 ~~buffer A~~ is full. After the peripheral device 50 occupies the bus 56 successfully, the bus interface circuit 54 drives

the buffer 58 ~~buffer A~~ to start outputting data to the bus 56, and simultaneously enables the buffer 59 ~~buffer B~~ to continuously receive data outputted from the controller 52. As mentioned above, the timing when the buffer 59 ~~buffer B~~ is fully loaded with data needs to be prior to the timing when the buffer 58 ~~buffer A~~ outputs all of its stored data for the sake of preventing the peripheral device 50 from entering the waiting state. That is, before the buffer 58 ~~buffer A~~ completes outputting all of its stored data, the buffer 59 ~~buffer B~~ must be fully loaded with data. When the buffer 59 ~~buffer B~~ is full, the remaining buffer 60 ~~buffer C~~ is still empty. Therefore, the bus interface circuit 54 can drive the buffer 60 ~~buffer C~~ to start receiving data outputted from the controller 52.

Please replace paragraph [0031] with the following amended paragraph:

[0031] When the buffer 58 ~~buffer A~~ completes outputting all of its stored data, the bus interface circuit 54 drives the buffer 59 ~~buffer B~~, whose capacity is full, to start outputting its stored data. Similarly, the timing when the buffer 60 ~~buffer C~~ is fully loaded with data needs to be prior to the timing when the buffer 59 ~~buffer B~~ outputs all of its stored data for the sake of preventing the peripheral device 50 from entering the waiting state. That is, before the buffer 59 ~~buffer B~~ completes outputting all of its stored data, the buffer 60 ~~buffer C~~ needs to be fully loaded with data. When the buffer 60 ~~buffer C~~ is full, the available buffer 58 ~~buffer A~~ is empty. Therefore, the bus interface circuit 54 can drive the buffer 58 ~~buffer A~~ to start receiving data outputted from the controller 52. When the buffer 59 ~~buffer B~~ completes outputting all of its stored data, the bus interface circuit 54 drives the buffer 60 ~~buffer C~~, whose capacity is full, to start outputting stored data. The identical operations mentioned above are repeated, and the buffers A, B, C can be used for receiving data delivered from the controller 52 and outputting stored data to the bus 56 sequentially. In the end, the same objective of outputting data to the bus 56 through the peripheral device 50 is achieved.